



POLITECHNIKA ŚLĄSKA
WYDZIAŁ INŻYNIERII BIOMEDYCZNEJ

Sprawozdanie z projektu z Inżynierii Programowania

Aplikacja webowa do komunikacji z lekarzem

Autorzy: D. Barański, K. Bielecka, M. Cebrat, P. Wróbel

Prowadząca projekt: Dr inż. Joanna Czajkowska

Zabrze, maj, 2019

Spis treści

1.	<i>Wstęp</i>	1
1.1	Cel projektu	1
1.2	Platforma projektowa	1
1.2.1	Angular	1
1.2.2	Firebase	2
1.2.3	JSON	2
1.3	Harmonogram prac	2
1.4	Przegląd rynku	2
1.4.1	Doktormed	2
1.4.2	Edoktor24	4
1.4.3	Grupa Luxmed	4
1.4.4	Kardiologonline	5
1.4.5	Medicover	5
2.	<i>System</i>	7
2.1	Architektura systemu	7
2.2	Funkcje wewnętrzne	8
2.3	Przypadki testowe	9
3.	<i>Wnioski</i>	13

Spis rysunków

1.1	Doktormed.	3
1.2	Edoktor24.	4
1.3	Grupa Luxmed.	4
1.4	Kardiologonline.	5
1.5	Medicover.	6
2.1	Podział modułu w Angular.	7
2.2	Podział modułu w Angular.	7
2.3	Architektura systemu.	8
2.4	Podział modułu w Angular.	8
2.5	Komponenty modułu autoryzacji.	9
2.6	Komponenty modułu czatu.	10
2.7	Komponenty modułu kliniki.	11

Spis tabel

1.1	Harmonogram prac.	3
-----	---------------------------	---

1. Wstęp

1.1 Cel projektu

Celem niniejszej pracy było utworzenie aplikacji webowej, wspomagającej kontakt pomiędzy lekarzem, a pacjentem. Docelowo, aplikacja powinna umożliwiać wymianę danych na bieżąco. Pacjent powinien mieć możliwość rozpoczęcia rozmowy z wszystkimi lekarzami dostępnymi on-line. Założono, iż aplikacja stanowi narzędzie, wspomagające pracę istniejącej już przychodni lekarskiej.

Osiągnięcie celu wymagało realizacji następujących etapów:

- przydziału zadań poszczególnym osobom w grupie,
- wyboru narzędzi,
- opracowania architektury bazy danych,
- stworzenia interfejsu użytkownika,
- implementacji poszczególnych funkcjonalności.

1.2 Platforma projektowa

1.2.1 Angular

Otwarty framework i platforma do tworzenia SPA, napisany w języku TypeScript i wspierany oraz rozwijany przez Google.

Elementy Angular:

- architektura MVW - aplikacje mogą być oparte o różne wzorce architektoniczne, z którymi Angular sobie poradzi;
- wstrzykiwanie zależności - wprowadzone przez nas funkcjonalności w kodzie stają się bardziej zautomatyzowane;
- moduły - moduł jest podstawowym nośnikiem danych tak jak klasa, jednak nie możemy tworzyć instancji modułu, opiera się on na nieco innej funkcjonalności;
- dwukierunkowe wiązanie danych - mechanizm dwukierunkowego wiązania danych zapewnia dynamiczną synchronizację danych między warstwą widoku, a warstwą modelu danych;

- nawigacja - możliwość przekierowywania, ingerowania w wyświetlanie widoku strony dla odpowiedniego adresu;
- filtrowanie danych - Angular oferuje wbudowane mechanizmy filtrowania, które poniekąd wyręczają deweloperów od pisania własnych funkcji filtrujących dane.

1.2.2 Firebase

Jest to BaaS odpowiedzialny za naszą architekturę backendową, za marketing, monitoring wydajności, zarządzanie uploadowanymi plikami, testowanie, modyfikowanie aplikacji i przechowywanie w bezpieczny sposób.

1.2.3 JSON

Jest prostym formatem wymiany danych. Jego definicja opiera się o podzbiór języka programowania JavaScript, Standard ECMA-262 3rd Edition - December 1999. JSON jest formatem tekstowym, całkowicie niezależnym od języków programowania, ale używa konwencji, które są znane programistom korzystającym z języków z rodziny C.

JSON powstał w oparciu o dwie struktury:

- Zbiór par nazwa/wartość. W różnych językach jest to implementowane jako obiekt, rekord, struktura, słownik, tabela hash, lista z kluczem, albo tabela asocjacyjna.
- Uporządkowana lista wartości. W większości języków implementuje się to za pomocą tabeli, wektora, listy, lub sekwencji.

Wspomniane struktury danych są uniwersalne. Prawie wszystkie nowoczesne języki programowania posługują się nim w tej lub innej formie. Ma to sens, by format danych, który jest przenośny pomiędzy różnymi językami programowania opierał swoją budowę na wspomnianych strukturach.

1.3 Harmonogram prac

Harmonogram prac został przedstawiony w tabeli (Tab. 1.1).

1.4 Przegląd rynku

1.4.1 Doktormed

Jest to platforma umożliwiająca zadawanie pytań do grona lekarzy w sposób anonimowy, koszt porady to 50 złotych i otrzymujemy odpowiedź od specjalisty z dziedziny zadanego przez nas pytania (Rys. 1.1).

Tab. 1.1: Harmonogram prac.

Termin	Wykonane prace
26.02.2019	Ustalenie tematu projektowego
12.03.2019	Zdefiniowanie problemu Wybór kierownika projektu Wybór platformy projektowej
26.03.2019	analiza rynku opracowanie architektury systemu zdefiniowanie funkcjonalności schematy użycia projekt GUI
09.04.2019	zamockowanie bazy danych Utworzenie ekranu logowania i logiki Utworzenie bazy danych leków wczytanie listy leków
07.05.2019	implementacja czatu wczytanie listy lekarzy/pacjentów
21.05.2019	Implementacja dodawania leków Walidacja logowania
04.06.2019	Dokumentacja Testowanie Dodanie historii chorób


doctormed

[Zadaj pytanie lekarzowi](#)
[Baza Wiedzy](#)
[Kontakt](#)

Lekarz online

Doctormed.pl to nowoczesny portal oferujący porady medyczne online. Bez kolejek, bez skrupowania.

Od początku pełna anonimowość to nasza główna zasada.

Pomagają lek. specjaliści

Zatrudniamy doświadczonych lekarzy różnych dziedzin. Dzięki swojej wiedzy i praktyce są w stanie pomóc w najróżniejszych przypadkach.

Udzielane odpowiedzi są pisane jasno, by były jak najbardziej pomocne.

Zalety doctormed

Przystępna cena	porada 50zł
Szybki czas odpowiedzi	nawet w 24h
Anonimowość	100%

Rys. 1.1: Doctormed.

1.4.2 Edoktor24

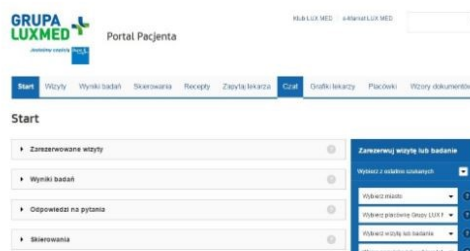


Rys. 1.2: Edoktor24.

Jest to platforma umożliwiająca zadanie pytań przez wybranie specjalisty zadanie pytania do którego jest możliwe dołączenie wyników badań a następnie po wniesieniu opłaty otrzymujemy odpowiedź w ciągu 24 godzin. Możliwe są także konsultacje telefoniczne lub wideo konsultacje z wybranym specjalistą (Rys. 1.2).

1.4.3 Grupa Luxmed

Godziny, w których można skorzystać z indywidualnej porady lekarza/położnej/pielęgniarki, widoczne są w zakładce CZAT. Aby rozpocząć rozmowę, wystarczy w godzinach pracy specjalisty medycznego wcisnąć przycisk ROZPOCZNIJ CZAT.



Rys. 1.3: Grupa Luxmed.

Jest to platforma umożliwiająca chat z pielęgniarzkami lub lekarzami wybranych specjalizacji w wyznaczonym czasie kiedy są dostępni (Rys. 1.3).

1.4.4 Kardiologonline



Rys. 1.4: Kardiologonline.

Jest to platforma umożliwiająca umówienie się na konsultację z wybranym specjalistą które odbędą się za pomocą menengera lub faceboka (Rys. 1.4).

1.4.5 Medicovert

Jest to czat dostępny za pomocą przeglądarki internetowej w którym pacjent może otrzymać odpowiedź na proste zadane pytanie do lekarza (Rys. 1.5).

Czaty z lekarzami



Czaty z lekarzami to szybki i prosty sposób na uzyskanie porady medycznej. Wystarczy dostęp do komputera lub telefonu z internetem, aby po zalogowaniu do Medicover Online skontaktować się z lekarzem Medicover.

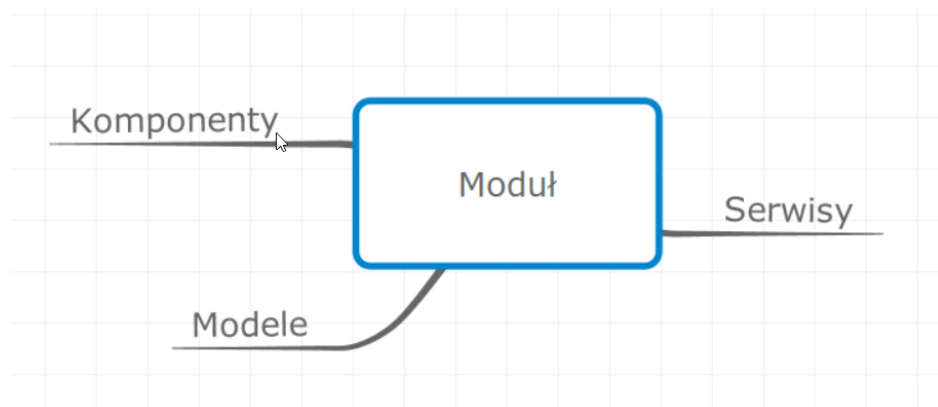
W trakcie czatów Pacjenci mogą otrzymać porady medyczne w przypadku:

- przeziębienia, gorączki, kaszlu
- bólów brzucha, wymiotów, gorączki
- infekcji układu moczowego
- wysypki
- profilaktyki zdrowia
- innych problemów zdrowotnych

Rys. 1.5: Medicover.

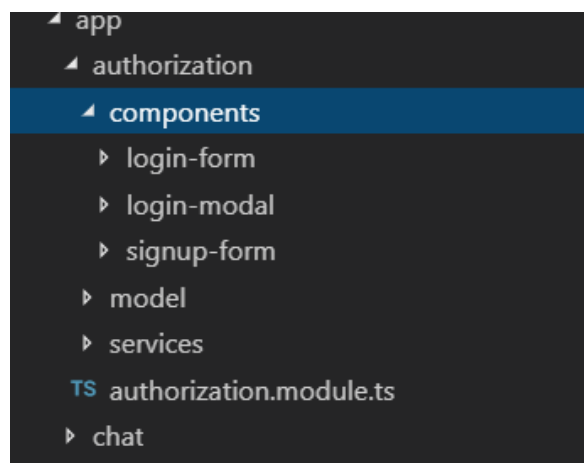
2. System

2.1 Architektura systemu



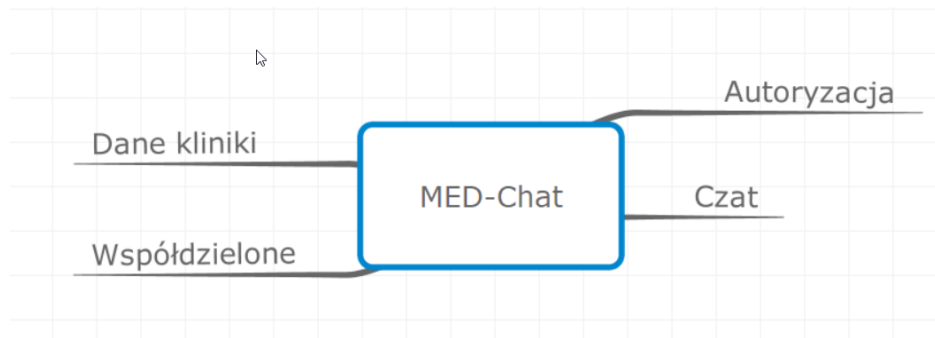
Rys. 2.1: Podział modułu w Angular.

Framework Angular wymaga pomaga utrzymać architekturę zgodną z praktykami dobrego programowania. Aplikacja dzieli się na moduły. Moduł jest odzwierciedleniem wzorca MVC (Rys. 2.1).



Rys. 2.2: Podział modułu w Angular.

Przykładowa implementacja modułu autoryzacji została przedstawiona na rysunku (Rys. 2.2).



Rys. 2.3: Architektura systemu.

Aplikacja została podzielona na 4 główne moduły, według funkcjonalności. Jest to główna architektura systemu (Rys. 2.3).



Rys. 2.4: Podział modułu w Angular.

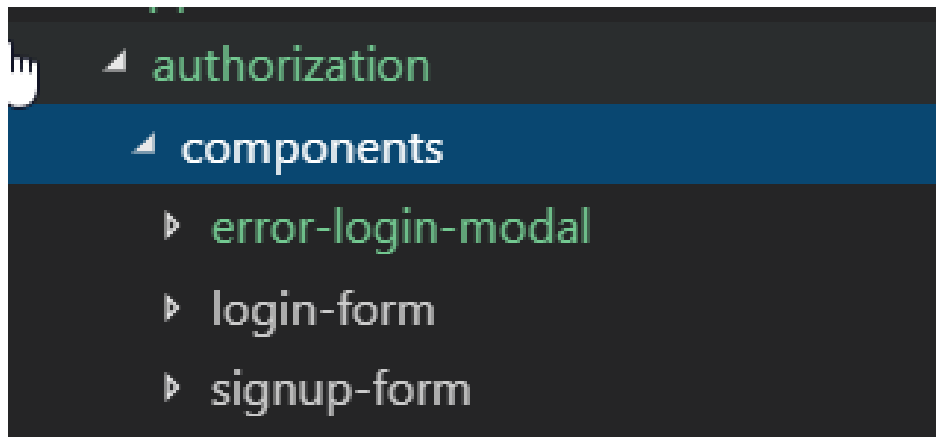
Aplikacja pobiera dane z bazy danych kliniki. Są to informacje o danych osobowych użytkownika, historia stosowanych leków oraz przebytych chorobach. Baza ta została napisana w formacie json i jest kompilowana za pośrednictwem biblioteki *json-server*, która jest na licencji *Open-source*. Można ją znaleźć w serwisie *Github* pod adresem <https://github.com/typicode/json-server>. Api do bazy danych zostało opublikowane za pomocą *Microsoft Azure*.

Do logowania oraz obsługi czatu użyto bazy danych *Firebase Cloud*. Dzięki niech możliwe jest pisanie i odbieranie wiadomości w czasie rzeczywistym. Płatność jest zgodna z użyciem.

Dokładny przepływ danych w aplikacji został przedstawiony na rysunku (Rys. 2.4).

2.2 Funkcje wewnętrzne

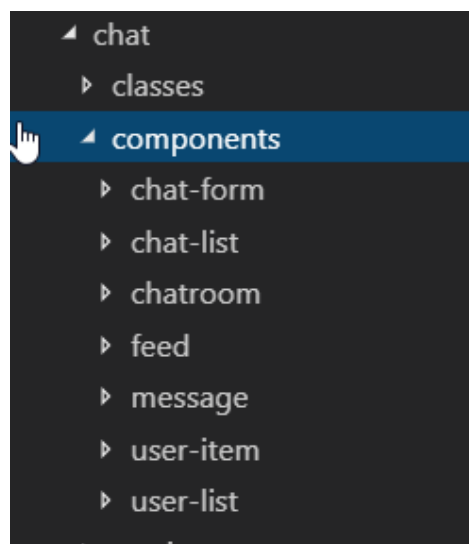
Zachowanie i podział na funkcje wewnętrzne aplikacji, najlepiej opisuje podział na komponenty wewnątrz modułów. W tej sekcji zostały opisane obowiązki poszczególnych komponentów.



Rys. 2.5: Komponenty modułu autoryzacji.

Komponenty modułu autoryzacji Moduł autoryzacji zawiera 4 komponenty (Rys. 2.5):

- *error-login-modal* - wyświetlanie błędu podczas próby logowania z błędnym loginem lub hasłem,
- *login-form* - formularz logowania,
- *signup-form* - formularz rejestracji,



Rys. 2.6: Komponenty modułu czatu.

Komponenty modułu czatu Moduł czatu zawiera 6 komponentów (Rys. 2.6):

- *chatroom* - główne okienko czatu, zawierające wszystkie pozostałe komponenty czatu;

- *chat-form* - formularz nowej wiadomości;
- *feed* - wyświetlenie wiadomości czatu;
- *message* - wyświetlenie pojedynczej wiadomości;
- *user-list* - wyświetlenie listy użytkowników;
- *user-item* - wyświetlenie pojedynczego użytkownika.

2.3 Przypadki testowe

Przetestowano aplikację przez opisywanie zachowania programu w formie historyjek użytkownika (ang. user stories). Zachowanie programu opisane jest w schemacie *Given, When, Then*. Skutkuje to otrzymaniem jednoznacznie sprawdzalnych przypadków testowych. Wykonane testy akceptacyjne złożone są z trzech sekcji.

- *Given* jest równoznaczna z warunkami początkowymi,
- *When* jest to akcja do wykonania,
- *Then* prezentuje oczekiwany rezultat.

Test 1

Given: widok logowania. Wprowadzenie loginu i błędnego hasła,
when: użytkownik naciska przycisk „Zaloguj”,
then: powinien wyświetlić się komunikat „Błąd logowania”.

Test 2

Given: widok logowania. Wprowadzenie poprawnego loginu i hasła,
when: użytkownik naciska przycisk „Zaloguj”,
then: powinien wyświetlić się główny widok czatu.

Test 3

Given: widok czatu. Wprowadzenie treści wiadomości,
when: użytkownik naciska przycisk „Wyślij”,
then: wiadomość powinna wyświetlić się w polu konwersacji.

Test 4

Given: obszar listy użytkowników,
when: użytkownik naciska element listy,
then: element powinien zostać podświetlony na niebiesko.

Test 5

Given: obszar listy użytkowników,
when: użytkownik naciska element listy, zawierającego imię i nazwisko,
then: w nagłówku konwersacji powinno wyświetlić się imię i nazwisko wybranej z listy osoby.

Test 6

Given: obszar listy leków,
when: użytkownik naciska element listy,
then: powinien zostać wyświetlony komunikat ze szczegółowymi informacjami.

Test 7

Given: komunikat ze komunikat ze szczegółowymi informacjami o leku,
when: użytkownik naciska przycisk „OK”,
then: komunikat przestaje być widoczny.

Test 8

Given: obszar historii chorób,
when: użytkownik naciska element listy,
then: element powinien zostać powiększony o dodatkowe informacje.

3. Wnioski

Cel pracy został osiągnięty – została utworzona aplikacja, umożliwiająca bezpośredni kontakt pomiędzy pacjentem a lekarzem, bez konieczności wcześniejszego planowania wizyty. Dzięki wykorzystaniu najnowszych technologii, utworzono usługę świadczenia opieki zdrowotnej na odległość. Do najważniejszych zalet powstałej aplikacji zalicza się:

- zmniejszenie kosztów opieki medycznej,
- dostęp do lekarzy bez długiego czasu oczekiwania na wizytę,
- kontakt z personelem medycznym w czasie urlopu,
- szybkie konsultacje lekarskie – możliwość poruszenia pilnych tematów, wymagających weryfikacji osoby doświadczonej,
- możliwość podglądu przepisanych leków,
- zredukowana liczba koniecznych dojazdów do placówki,
- szybki i łatwy dostęp do historii chorób zarówno ze strony lekarza, jak i od strony użytkownika.

W przyszłości, aplikacja może zostać rozbudowana o kolejne moduły, zwiększające funkcjonalność. Można stworzyć przydział danego lekarza do specjalizacji, możliwość wysyłania załączników lub przeprowadzania rozmów wideo.